



## **Slow Computing Simulation of Bio-plausible Control**

**by Alec Koppel, Vishnu Ganesan, Alma Wickenden, William Nothwang,  
Robert Proie, and Brian Sadler**

**ARL-TR-5959**

**March 2012**

## **NOTICES**

### **Disclaimers**

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

# **Army Research Laboratory**

Adelphi, MD 20783-1197

---

---

**ARL-TR-5959**

**March 2012**

---

## **Slow Computing Simulation of Bio-plausible Control**

**Alec Koppel, Vishnu Ganesan, Alma Wickenden, William Nothwang,  
Robert Proie, and Brian Sadler  
Sensor and Electron Devices Directorate, ARL**

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>				
1. REPORT DATE (DD-MM-YYYY) March 2012		2. REPORT TYPE Final		3. DATES COVERED (From - To) June 2011
4. TITLE AND SUBTITLE Slow Computing Simulation of Bio-plausible Control			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Alec Koppel, Vishnu Ganesan, Alma Wickenden, William Nothwang, Robert Proie, and Brian Sadler			5d. PROJECT NUMBER	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-SER-L 2800 Powder Mill Road Adelphi, MD 20783-1197			8. PERFORMING ORGANIZATION REPORT NUMBER  ARL-TR-5959	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT <p>In order to implement control methods on low power, size-constrained systems, significant departures from standard approaches are needed. We look to biology for inspiration and focus on the problem of attitude stabilization through visual input alone. Bio-plausibility broadly refers to the ability of an operation to be carried out on a neural substrate or a parallel asynchronously operating computation scheme. One method of realizing such an asynchronous, parallel computing capability in power-constrained systems is by making use of low-throughput, energy-efficient computing elements; this is referred to as a "slow computing" architecture. In this work, we evaluate the capacity of bio-plausible control laws for attitude stabilization on a slow computing architecture. The Gillespie Stochastic Simulation Algorithm (SSA) from chemical kinetics is modified to simulate a slow computing system. We approximate how sparse the system's visual information may be while remaining controllable and producing attitude stabilization. Gaussian noise is added to the simulation of visual input values, and the interaction of noise and sparsity is explored. The bio-plausible approach to stabilization is intended to replace more power-consumptive methods. In simulation, low power processing of visual input data produced effective attitude stabilization for a range of noise and sparsity.</p>				
15. SUBJECT TERMS Control theory, robotics, slow computing, bio-plausible				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT  UU	18. NUMBER OF PAGES  32
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified		
				19b. TELEPHONE NUMBER (Include area code) (301) 394-2054

---

## Contents

---

<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>v</b>
<b>Acknowledgments</b>	<b>vi</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Background</b>	<b>2</b>
2.1 Stochastic Simulation .....	2
2.2 Attitude Motion .....	4
2.3 Sensor Model and Visual Attitude Stabilization .....	5
<b>3. Experiment</b>	<b>6</b>
3.1 Angular Velocity Estimation.....	6
3.1.1 Traditional Computation Method .....	6
3.1.2 Bio-plausible Computation Method .....	6
3.2 Control Law and Discretization of the Visual Field.....	7
3.3 Adapting the Stochastic Simulation Algorithm.....	8
3.3.1 Sparsity .....	10
<b>4. Results and Discussion</b>	<b>10</b>
<b>5. Conclusions</b>	<b>15</b>
<b>6. Further Directions</b>	<b>16</b>
6.1 Improving the Physics .....	16
6.2 Contrast Relationship .....	16
6.2.1 A Probability Density Function for $\mathfrak{D}$ .....	17
6.2.2 Kernel Smoothing.....	17
6.2.3 Network Architecture .....	17
6.2.4 Learning Sensor Orientation .....	17
6.2.5 Broadening the Control Domain .....	18
6.2.6 Hardware Implementation .....	18

<b>7. References</b>	<b>20</b>
<b>List of Symbols, Abbreviations, and Acronyms</b>	<b>22</b>
<b>Distribution List</b>	<b>23</b>

---

## List of Figures

---

Figure 1. Example calculation of a stoichiometric matrix.....	2
Figure 2. Yaw, pitch, and roll constitute the orientation or attitude. ....	4
Figure 3. Example simulation of compound fly vision in GUF. ....	10
Figure 4. (a) Attitude stabilization using least squares estimation of angular velocity. (b) Standard deviation in pixel luminosity for Logitech webcam observing a static scene. ...	11
Figure 5. Bio-plausible attitude stabilization in the idealized, noisy, and sparse cases, respectively. ....	11
Figure 6. Examples of varying levels of information sparsity and propensities for computational elements to provide visual control information. ....	13
Figure 7. Map of stability control using the Slow Computing algorithm, as a function of noise and information sparsity. Note that noisier and sparser visual information degrades the control signal to a point where the system cannot stabilize. ....	15
Figure 8. Roadmap of design efforts towards control of an insect-scale autonomous robotic platform. ....	19

---

## List of Tables

---

Table 1. Gillespie stochastic simulation algorithm for chemical kinetics. ....	3
Table 2. Slow Computing SSA. ....	9
Table 3. Different parameters that have an effect on attitude stabilization. ....	16

---

## Acknowledgments

---

We would like to acknowledge the helpful guidance of Prof. Richard Murray and Mr. Shuo Han of the California Institute of Technology. The current simulation was based on a simulation framework that was provided by the Murray group, and the Grand Unified Fly (GUF) simulation framework developed by Dr. Andrew Straw.



---

## 1. Introduction

---

Control systems design focuses on the use of a driving term to elicit a particular behavior in a dynamical system. The control term that drives the system response has conventionally been designed with the assumption that robust computer power is available and the size of such a computer is not a limiting factor. In this context, relatively complicated operations are not an impediment to effective control. Here, however, we focus on the problem of stabilization of a small, power-constrained platform. Traditional localization and stabilization methods are based on global positioning system (GPS), real-time kinetic (RTK), or other computationally intensive methods for obtaining information about a system's environment. Visual stabilization refers to platform stabilization based on visual input. Traditional control methods for visual stabilization are too power intensive to be miniaturized effectively.

There is a need to design and implement small-scale, low power robotic systems with multisensory input modalities to enable the next generation of autonomous technology. Focusing on low computing power and miniaturization motivates the need for unconventional ways to approach control. Small organisms, especially insects, are marvels of control. They are able to process huge quantities of sensory input and actuate a response on a relatively small timescale. For this reason, researchers have begun using biological principles as guidance on how to approach power and size-constrained control problems. Engineers look to insects, especially the fruit fly *Drosophila melanogaster*, for inspiration on how to achieve effective control of autonomous systems on small, low power platforms. One could imagine a threatened fly leaping into the air haphazardly and within fractions of a second, effectively pursuing a correct flight direction.

*Bio-plausibility* refers to a loose collection of engineering principles that seek to confine design research to that which could occur in a biological process (such as a neural network). Control theory inspired by the principles of biological systems has recently made significant advancements due to the work of Dickson, et al. (1), Epstein et al. (2), Fry et al. (3), and Han et al. (4). What can be considered bio-plausible control theory is loosely defined. Operations must not be "too computationally expensive," in that formulas should be, at worst, locally nonlinear functions that can be computed on parallel asynchronous units. Each operation is then integrated systemically through a linear operation. For example, a control law that requires taking a matrix inverse would not be considered bio-plausible; reducing such a control law to a bilinear form would be an improvement. Traditional computation methods can be simplified to reduce the power cost. This savings, however, comes with a nontrivial loss of accuracy. Given that biological systems do not operate via computationally intensive and synchronized processes, we expect this loss of fidelity. In control systems, nature compensates by increasing the number of

sensor inputs, parallelizing sensor processing, and rapidly responding. Thus, the parallel nature of the processors is key to eliciting a low power response.

## 2. Background

### 2.1 Stochastic Simulation

Before well-behaved micro-autonomous robots can be designed, the fidelity and stability of these bio-plausible control laws must be understood on a slow computing architecture. Slow computing, defined as asynchronous parallel processing using low-throughput, energy-efficient elements, can be simulated via the framework of stochastic chemical kinetics famously pioneered by Gillespie (6) with some modifications made in order to apply it to this control theoretic context. Bio-plausible control has been developed and simulated in series in recent works by Han et al. (4) and Conroy et al. (7); the Monte Carlo approach that we apply to this parallel computing problem will provide new insight to the control system's behavior in the face of sparse and asynchronous information accumulation.

We briefly review the Gillespie Stochastic Simulation Algorithm (SSA) to provide context for its applicability to bio-plausible control theory. The SSA is proven to be a physically exact realization of a *well-stirred*, chemically reacting system on the scale of individual species  $X_1, \dots, X_n$ . Suppose  $n$  is the number of species and  $m$  is the number of reactions for some reaction mechanism. Let  $X \in \mathbb{R}^n$  be the state vector whose components are the quantities of chemical species at a given time. Any collection of reactions can be represented as a matrix of coefficients for the reactants and products called the *stoichiometric matrix*  $U \in \mathbb{R}^{n \times m}$ . For example, the reactions  $r_1: X_1 \rightarrow 2X_1$  and  $r_2: X_1 + X_2 \rightarrow 2X_2$  give rise to the matrix representation  $U = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$ . Bause and Kritzinger (8) give a more detailed treatment of graphical and matrix representations of reaction mechanisms, as illustrated in figure 1.

Post		Pre		$U$
$x_1 \begin{bmatrix} 2 & 0 \end{bmatrix}$	$-$	$x_1 \begin{bmatrix} 1 & 1 \end{bmatrix}$	$=$	$x_1 \begin{bmatrix} 1 & -1 \end{bmatrix}$
$x_2 \begin{bmatrix} 0 & 2 \end{bmatrix}$		$x_2 \begin{bmatrix} 0 & 1 \end{bmatrix}$		$x_2 \begin{bmatrix} 0 & 1 \end{bmatrix}$
Product		Reactants		Stoichiometric
coefficients		coefficients		matrix
matrix		matrix		

Figure 1. Example calculation of a stoichiometric matrix.

We can turn this collection of chemical reactions into a dynamical system, as illustrated in table 1. We assume each chemical reaction contains an inherent reaction rate constant  $c_i$  and the reactions occur according to some rate functions. We call  $h(c_i, X)$  the rate function for reaction

$r_i$  and denote the combined rate function  $h(X) = \sum_i h(c_i, X)$ . The *Law of Mass Action* is a common choice of reaction rate functions.

Table 1. Gillespie SSA for chemical kinetics.

Gillespie SSA	
1.	Initialize the system at initial state $X_0$ (which typically corresponds to $t = 0$ ).
2.	For each reaction, calculate $h(c_i, X)$ , the reaction rate function.
3.	Compute the system wide rate $h(X) = \sum_i^m h(c_i, X)$ .
4.	Compute the delay time until the next reaction: simulate a sample value $s$ from the exponential distribution with a rate $h(X)$ .
5.	Set the current time to $t + s$ and call it $t = 0$ .
6.	Choose the next reaction: Simulate an index $j$ according to the probability distribution given by $\frac{h(c_i, X)}{h(X)}$ .
7.	Update the quantity of species $j$ by the appropriate column of the stoichiometric matrix $U_j$ .
8.	Save $X$ and $t$ . If $t < T_{max}$ , return to step two.

The chemical reactions are assumed to occur independently of one another due to the well-stirred assumption. The correct stochastic process to describe the evolution of chemical quantities (which are discrete positive integers) is the Poisson Process. Consequently, the waiting time between events are exponentially distributed with a rate given by the combined rate function  $h(X)$ . If the present time is  $t$ , the waiting time until the next event is determined via a sample value  $s$  from the exponential distribution with rate  $h(X)$ . The algorithm chooses which reaction will fire at this time by choosing an index  $j$  from  $\{1, \dots, m\}$  according to the probability distribution generated by the reaction rate functions, normalized by the combined rate  $h(X)$ :  $\frac{h(c_i, X)}{h(X)}$ . The system then updates the state vector  $X$  according to the stoichiometric matrix:  $X + U_j$  (where  $U_j$  is the  $j$ th column of the stoichiometric matrix) and repeats until some assigned stopping time. This stochastic update rule gives an exact realization of the Chemical Master Equation (CME), a discrete stochastic partial differential equation that models this chemical reaction mechanism. The CME is given below:

$$\frac{\partial P(X, t | X_0, t_0)}{\partial t} = \sum_{i=1}^m h(c_i, X - U^i) P(X - U^i) - h(c_i, X) P(X) \quad (1)$$

Here  $P(X, t)$  denotes the probability that  $X_t = X$  at time  $t$  with  $h$  the reaction rate function,  $c_i$  the  $j$ th reaction rate constant, and  $U_i$  the  $i$ th column of the stoichiometric matrix as before. This dynamical system is a continuous-time Markov Chain with a discrete state space. See the works of Gillespie (6), Higham (9), and Khammash (10) for additional details. This manner of stochastic simulation will be essential to simulating an asynchronous parallel computing structure.

## 2.2 Attitude Motion

Our goal is visual attitude stabilization of the control system. Attitude refers to the orientation in three-dimensional (3-D) space, as illustrated in figure 2. Some preliminaries are necessary to establish the formal language of orientation. Elements of the Lie group  $SO(3)$  are rotation matrices with the two key properties: for every  $R \in SO(3)$ ,  $R^T R = I = R R^T$  and  $\det(R) = 1$ , where  $I$  is the identity in  $SO(3)$ . Thus, elements of  $SO(3)$  are orthogonal. For example, a rotation matrix that rotates vectors about an angle  $\theta$  about the vertical axis  $z$  takes the form

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Matrices in  $SO(3)$  describe all possible orientations of a rigid body and rotation matrix products represent composition of rotations. The tangent space to  $SO(3)$  for a particular element  $R$  is the set of *all possible angular velocities* the rigid body can attain at that attitude. Consider the tangent vector space to  $I \in SO(3)$ , which is the Lie algebra  $TSO(3)$ . Elements of  $TSO(3)$  are skew-symmetric matrices corresponding to vectors in  $\mathbb{R}^3$ : for a given angular velocity  $\omega \in \mathbb{R}^3$  of the form  $[\omega_1 \ \omega_2 \ \omega_3]^T$ , the corresponding skew-symmetric matrix  $\omega_\times \in TSO(3)$  takes the form

$$\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & \omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (3)$$

which is a *matrix representation of the cross-product*. Thus,  $TSO(3)$  is isomorphic to  $\mathbb{R}^3$ . In some literature,  $TSO(3)$  is denoted as  $\mathfrak{so}(3)$ . For a more complete background on this material, see Baker (11), Chaturvedi, Sanyal, and McClamroch (12), Diebel (13), Murray, Li, and Sastry (14), and Stuelpnagel (15).

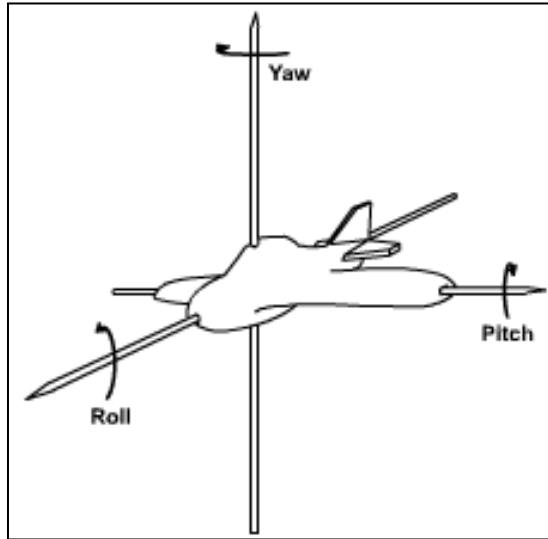


Figure 2. Yaw, pitch, and roll constitute the orientation or attitude.

Equipped with these tools, we can fix a world frame and describe the rotational motion of a fully actuated rigid body through a second-order system:

$$\begin{cases} \dot{R} = R\omega_{\times} \\ \mathbb{I}\dot{\omega} = (\mathbb{I}\omega) \times \omega + \tau \end{cases} \quad (4)$$

Here  $R \in SO(3)$  is the body attitude with respect to a fixed world frame;  $\omega \in \mathbb{R}^3$  and  $\omega_{\times} \in TSO(3)$  are the angular velocity and its associated skew-symmetric matrix, respectively;  $\mathbb{I}$  is the  $3 \times 3$  symmetric angular inertia matrix; and  $\tau$  is the input torque. Note that in our simulations in the present work,  $\mathbb{I}$  is set to the identity matrix.

### 2.3 Sensor Model and Visual Attitude Stabilization

We assume at each time and orientation, a collection of sensors output a visual field. This discrete visual field is equally applicable to standard cameras, catadioptric cameras, or the compound vision of a fruit fly. Each visual sensor maps the time and a vector on the unit sphere  $\mathbb{S}^2 \subset \mathbb{R}^3$  (thought of as an arrow attached to a sphere of radius 1 in  $\mathbb{R}^3$ ) to an observed luminance through a function  $y: \mathbb{S}^2 \times \mathbb{R} \rightarrow \mathbb{R}$  defined on the unit sphere:

$$y(s, t) \equiv m(R(t)s) \quad (5)$$

where  $m$  is some known map of the orientation, environment, and time to the visual input  $y$ . Therefore, the entire compound eye of the fly is the set of ommatidia functions  $y_i, 1 \leq i \leq 1398$  for 1398 ommatidia, as in Han et al. (4). The bio-plausible control laws in the following discussion have been derived using a few key assumptions: occlusions are ignored, the set of discrete luminance functions  $\{y_i\}$  is sampled from a continuous visual field, and both  $y_i$  and  $\dot{y}_i$  can be observed.

The visual input  $y \in [0,1]$  is a raw luminance. With the visual field, angular velocity  $\omega$ , and orientation  $s$  we are able to obtain the optic flow  $\dot{y}$ , an approximation of real-time motion through variable image luminosity per Fry et al. (3). This optic flow equation is simplified to only incorporate rotational motion.

$$\dot{y}(s, t) = (s \times \nabla_s y(s, t))^T \omega(t) \quad (6)$$

The superscript  $^T$  refers to the transpose and the subscript under the gradient operator refers to its evaluation at a particular orientation  $s \in \mathbb{S}^2$ . In the spirit of Dickson et al. (1) and Han et al (4), we may simplify this expression by defining a linear differential operator  $S: \mathcal{C}^1(\mathbb{S}^2, \mathbb{R}) \rightarrow \mathfrak{X}(\mathbb{S}^2)$ :

$$Sy \equiv s \times \nabla_s y \quad (7)$$

where  $\mathcal{C}^1(\mathbb{S}^2, \mathbb{R})$  refers to the space of continuous functions from  $\mathbb{S}^2$  to  $\mathbb{R}$  and  $\mathfrak{X}(\mathbb{S}^2)$  is the tangent vector field to the unit sphere. Then the rotational optic flow relationship previously presented takes a cleaner form after dropping the parameterization by  $s$  and  $t$ :

$$\dot{y} = (Sy)^T \omega \quad (8)$$

The previous equation has a particularly nice geometric interpretation: find a vector normal to the present orientation and change in luminance evaluated at this orientation, and then take the inner product with this normal vector and the system's present angular velocity to produce the scalar change in luminance.

We predefine a goal image  $g: \mathbb{S}^2 \rightarrow \mathbb{R}$ , which is a particular value of  $y$  at a certain goal orientation  $R^g$ . The problem of *visual attitude stabilization* succinctly is stated as choosing the input torque  $\tau$  such that  $y \rightarrow g$  so that the system orientation  $R \rightarrow R^g$  and the angular velocity approaches zero. We aim for our system to converge its visual input to the goal image. For computational expediency, we program rotations via *quaternions*, an isomorphic description of rotation to  $SO(3)$ . Adams (16) provides information on vector fields over spheres. Chaturvedi (12) provides an extensive treatment of attitude stabilization. Diebel (13), Stuelpnagel (15), and Tayebi (17) provide insight on quaternions in this context.

---

### 3. Experiment

---

#### 3.1 Angular Velocity Estimation

##### 3.1.1 Traditional Computation Method

$y$  and  $\dot{y}$  are very noisy. Signal processing considerations tell us that the noise in  $y$  is insignificant compared to that of  $\dot{y}$ . Therefore, we assume a statistical model of the form  $\dot{y} = S(y)^* \omega + \varepsilon$ , where the error term  $\varepsilon$  is assumed to be independent and identically distributed (18). The *least-squares estimate* of angular velocity may be considered a traditional calculation method:

$$\hat{\omega}_{LS} \equiv \langle (Sy)(Sy)^T \rangle^{-1} \langle \dot{y} (Sy) \rangle \quad (9)$$

where the functional  $\langle f \rangle$  represents the Lebesgue integral over the sphere with respect to the rotational-invariant measure  $dS$ :

$$\langle f \rangle = \int_{\mathbb{S}^2} f dS \quad (10)$$

The vector product  $(Sy)(Sy)^T$  is sometimes referred to as the outer product and produces a matrix. Note that the computation of  $\hat{\omega}_{LS}$  involves the Fischer Information Matrix for  $\omega$ :

$$\mathcal{I}[\omega] \equiv \langle (Sy)(Sy)^T \rangle \quad (11)$$

Calculating  $\hat{\omega}_{LS}$  requires inverting  $\mathcal{I}[\omega]$ , which is a global nonlinear operation. Also,  $\mathcal{I}[\omega]$  may not be invertible in which case a pseudo inverse gives the best least-squares estimate.

##### 3.1.2 Bio-plausible Computation Method

$\hat{\omega}_{LS}$  is considered a “traditional” control law because of its large computational overhead and its inability to be parallelized. The matrix inverse (or pseudo inverse) along with other nonlinear

operations cannot be decomposed component-wise for parallel execution. There is no discernible linear relationship between  $\omega$ ,  $y$ , and  $\dot{y}$ ; a simpler relationship put forth by Dickson et al. (1) and Han et al. (4) is bilinear. The angular velocity  $\omega$  may be estimated via a *bilinear form* in  $y$  and  $\dot{y}$  that essentially replaces the  $J[\omega]^{-1}$  with a scalar  $c > 0$ , giving

$$\hat{\omega}_{BL} \equiv c\langle\dot{y}(Sy)\rangle \quad (12)$$

It can further be shown that  $\hat{\omega}_{BL}$  is an unbiased estimator of  $\omega$ ,  $c$  is the average image contrast:  $c = (\{E_m\|\nabla y\|_2^2\})^{-1}$ , and  $\hat{\omega}_{BL}$  is a skew-symmetric bilinear operator (18). Despite potential arbitrary scale inaccuracies related to the brightness of the environment,  $\hat{\omega}_{BL}$  remains useful for control systems design. Since the constant  $c$  is folded into the gain terms in the control law presented in the following section, we focus attention on the gains.

### 3.2 Control Law and Discretization of the Visual Field

The problem of visual attitude stabilization amounts to minimizing an error function that represents how “far” away the present visual input is from the goal image. The error function we chose to minimize is  $J(R) = \frac{1}{2}\|g - y\|_2^2$ . From Censi et al. (18), we know the gradient flow that minimizes this cost function is  $\langle g(Sy)\rangle$ . The following control law is a proportional/derivative (PD) controller with  $\langle g(Sy)\rangle$  as the proportional part and  $\langle\dot{y}(Sy)\rangle$  as the derivative part. The damping term  $\langle\dot{y}(Sy)\rangle$  is necessary to choose the torque such that  $\omega$  is driven to zero near the goal image.

$$\tau = k_p\langle g(Sy)\rangle - k_d\langle\dot{y}(Sy)\rangle \quad (13)$$

This control makes  $R = R^g$ ,  $\omega = 0$  *locally asymptotically stable*. As previously defined,  $\langle\dot{y}(Sy)\rangle = \hat{\omega}_{BL}$  and  $\nabla R = \langle g(Sy)\rangle$ . Therefore, the above control simplifies to  $\tau = k_p\nabla R - k_d\hat{\omega}_{BL}$  (18).

Implementing this control law within our sensor framework will require discretizing the visual field because we have a finite integer number of visual inputs. We denote the discretization of  $Sy$ :

$$Discretize[Sy] = Ay \quad (14)$$

$A$  is an  $n \times n \times 3$  tensor in full generality, where the last dimension  $k$  corresponds to the axis of rotation, of which there are three such axes. Entries of  $A$  are denoted  $A_{ijk}$  and  $A_{ij} \in \mathbb{R}^3$  (4). Once discretized, each visual sensor’s luminance value contributes to the torque relationship shown earlier. The discrete approximation to  $S$  is comparable to the Sobel operator used for edge detection in image processing; however, this approach does not assume a uniformly sampled visual field. This allows an arbitrary distribution of ommatidia sampling, which is ideal for a stochastic parallel computing architecture. Tailoring  $S$  to the fly visual field is done according to the  $A$  tensor:

$$A_{ij} = -q'(\alpha_{ij}) \cdot (s_i \times s_j) / \sin(\alpha_{ij}) \quad (15)$$

Here  $q$  is a smooth kernel (Gaussian is used as default),  $q'$  is its derivative, and  $\alpha_{ij} \equiv \cos^{-1}(s_i \cdot s_j)$  is the angle formed by  $s_i$  and  $s_j$ . More information on kernel smoothing may be found in works by Simoncelli, Adelson, and Heeger (19) and Wand and Jones (20). Also, by convention we set  $A_{ii} = 0$  to avoid any indeterminate entries of  $A$ . The kernel may be chosen to be localized and sparsely approximates  $A$ . We obtain optic flow in this discrete setup via a simple rate of change formula  $\dot{y} = \frac{y(t) - y(t - \Delta t)}{\Delta t}$ . With this notation, the control law is reformulated:

$$\tau_k = \sum_{i,j} y_i A_{ij} (k_p g_j - k_d \dot{y}_j) \quad (16)$$

It remains to parallelize the computation of  $\hat{\omega}_{BL}$  and  $\nabla R$ . Due to physical considerations of potential circuit test beds, we break down the previous computation according to the number of sensors  $n$ . For a pure focus on optimizing computation time of  $\hat{\omega}_{BL}$  and  $\nabla R$ , one might instead choose to compute each multiplication term in the dot product above in parallel for a total of  $n^2$  parallel computing components. In our implementation, each parallel processor computes  $\dot{y}_j(A_{jk} \cdot y)$ , a component of  $\hat{\omega}_{BL}$ , such that

$$\hat{\omega}_{BL,k} = \sum_{j=1}^n \dot{y}_j(A_{jk} \cdot y) \quad (17)$$

where  $A_{jk}$  is the  $j$ th row vector of  $A$  about axis  $k$ . Note that  $\hat{\omega}_{BL} \in \mathbb{R}^3$  so its components are computed for  $k = 1, 2, 3$  corresponding to the x, y, and z axes, respectively. An almost identical component-wise decomposition of  $\nabla R$  is implemented:

$$\nabla R_k = \sum_{j=1}^n g_j(A_{jk} \cdot y) \quad (18)$$

where the computation of  $g_j(A_{jk} \cdot y)$  also occurs at each parallel processor.

### 3.3 Adapting the Stochastic Simulation Algorithm

We simulate a slow computing architecture by adapting the SSA as follows: the state of the system is a computational record of the number of updates that occur at each individual visual component. Each chemical species represents a computing element; the reactions in the system record the number of computations in the slow computing architecture.

The first-order chemical reaction used to record the number of computations for a particular visual component is  $X_i \rightarrow 2X_i$ . Thus, the system-wide stoichiometric matrix is  $I_n$ , the  $n \times n$  identity matrix. We add an additional step: the computing of a portion of  $\hat{\omega}_{BL}$  and  $\nabla R$ , which occurs after the saving of the computing record of each visual component according to the stoichiometric matrix and the event time according to the Poisson Process. The system-wide control input occurs according to a Decision Time rate  $\mathfrak{D}$  in which the state vector defaults back to an initial state  $X_0$ . That is, at  $t = \mathfrak{D}$ , the system computes  $\hat{\omega}_{BL}$  and  $\nabla R$  by summing up all the



accumulated visual information  $\dot{y}_j(A_{jk} \cdot y)$  and  $g_j(A_{jk} \cdot y)$ . It then outputs a control torque, and sets  $X^{(\mathfrak{D})} = X^{(0)}$

The Law of Mass Action is not an appropriate method to choose reaction rate functions in this context. We want our reaction rate functions to be inversely proportional to the number of updates that occur: once a particular component of the visual field has been updated, it should become less likely to be refreshed in the near future. Here we propose the *Inverse Concentration Method* for the reaction rate functions:  $h(c_i, X) = \frac{c_i}{X_i}$ , where  $X_i$  is the number of computations for the  $i$ th visual component and as usual  $c_i$  is the reaction rate constant. This method is only valid for nonzero values of  $X_i$ , but will not be of concern here because the default initial state is set to  $X_0 = (1, \dots, 1)$  and the number of computations is increasing. This inverse relationship can of course be dominated by reaction rate constants to increase the update propensity of a particular region of the visual field relative to other regions. The reaction rate constants are chosen according to the physical constraints of the system. Table 2 details the slow computing SSA.

Table 2. Slow Computing SSA.

Slow Computing SSA
1. Initialize the system at initial state $X_0 = (1, \dots, 1)$ .
2. For each reaction, calculate $h(c_i, X) = \frac{c_i}{X_i}$ , the reaction rate function
3. Compute the system-wide rate $h(X) = \sum_i^n h(c_i, X)$ .
4. Compute the delay time until the next reaction: simulate a sample value $s$ from the exponential distribution with combined rate $h(X)$ .
5. Set the current time to $t+s$ and call it $t$ .
6. Choose the next index to be updated: simulate a sample index $j$ according to the probability distribution given by $\frac{h(c_i, X)}{h(X)}$ for $i=1, \dots, n$ .
7. Add 1 to the number of computations for component $j$ .
8. Compute $\dot{y}_j(A_j \cdot y)$ and $g_j(A_{jk} \cdot y)$ the $j$ th components of $\hat{\omega}_{BL}$ and $\nabla R$ , respectively.
9. Save $X$ and $t$ .
10. If $t < \mathfrak{D}$ , return to step two.
11. If $t = \mathfrak{D}$ , compute $\hat{\omega}_{BL,k} = \sum_j \dot{y}_j(A_j \cdot y)$ , $\nabla R_k = \sum_{j=1}^n g_j(A_{jk} \cdot y)$ , and the corresponding control torque $\tau_k = -k_p \nabla R_k - k_d \hat{\omega}_{BL}$ .
12. Return to step 1 and repeat until the set stopping time in the Grand Unified Fly (GUF).

This Monte Carlo approach to simulating parallel bio-plausible control will inform future design of parallel circuitry on field-programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs). Our simulations are implemented on the Grand Unified Fly (GUF) component *fsee*, a fly vision simulation environment illustrated in figure 3.

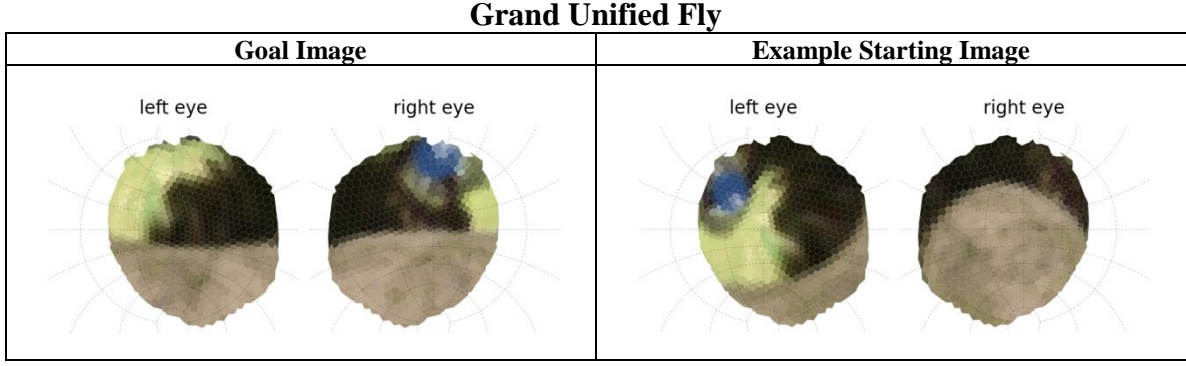


Figure 3. Example simulation of compound fly vision in GUF.

### 3.3.1 Sparsity

The decision time rate  $\mathfrak{D}$ , is related to what we call the sparsity of the system. A system with a sparsity of 0.5 will make a decision when approximately 50% of the computational elements have responded. As the decision time rate  $\mathfrak{D}$  decreases, the system makes decisions with data from fewer and fewer computational elements. We call this increasing the sparsity. We assume that in slow computing systems there is discretization at the level of decision making, that is, decisions are made discrete time points. A decision could be thought of as wing motor neuron firing. Sparsity could be thought of as the information availability at that time point.

## 4. Results and Discussion

A comparison of the bio-plausible and traditional angular velocity estimates is needed to establish the efficacy of  $\hat{\omega}_{BL}$  as a parallelizable replacement for  $\hat{\omega}_{LS}$ . It was anticipated that the more complete visual data set considered by the least-squares estimation in its statistical parameter estimation would provide increased accuracy for  $\hat{\omega}_{LS}$  over  $\hat{\omega}_{BL}$ . This expectation was confirmed by the nominally 0.2 s to stabilization of  $\hat{\omega}_{LS}$ , in the simulation shown in figure 4a, versus an approximately 0.5-s stabilization of  $\hat{\omega}_{BL}$ , shown in figure 5a, under identical conditions. The higher frequency oscillation observed in figure 4a may be an indicator of greater control precision. As a comparison, one could imagine a highly precise analytical balance ( $\hat{\omega}_{LS}$ ) compared with a triple beam balance ( $\hat{\omega}_{BL}$ ): the analytical balance will fluctuate much more. Nevertheless, under identical simulation conditions save the gains,  $\hat{\omega}_{BL}$  brings the system to attitude stabilization on a near identical time scale within a comparable band of geodesic distance from the goal image (near zero) to  $\hat{\omega}_{LS}$ . Geodesic distance can be thought of as arc length across a sphere; the geodesic degree follows naturally. When geodesic distance and degrees approach zero, the instantaneous image becomes the goal image.

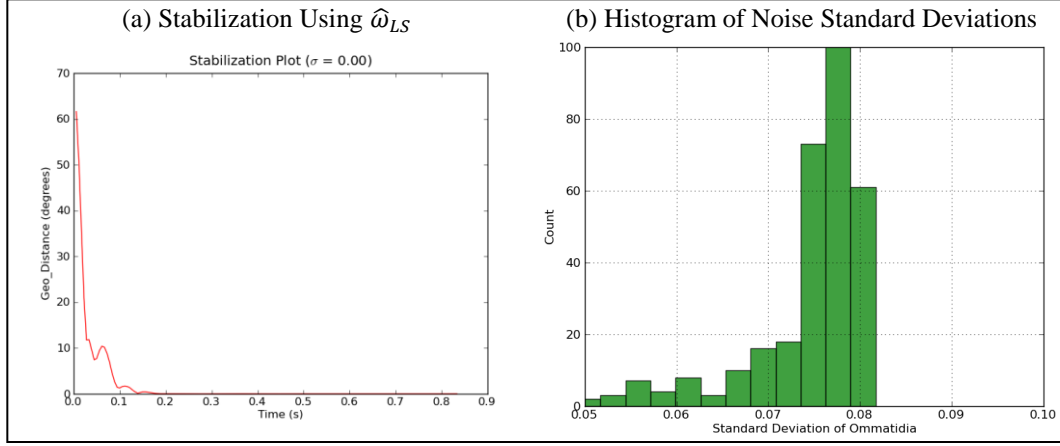


Figure 4. (a) Attitude stabilization using least squares estimation of angular velocity. (b) Standard deviation in pixel luminosity for Logitech webcam observing a static scene.

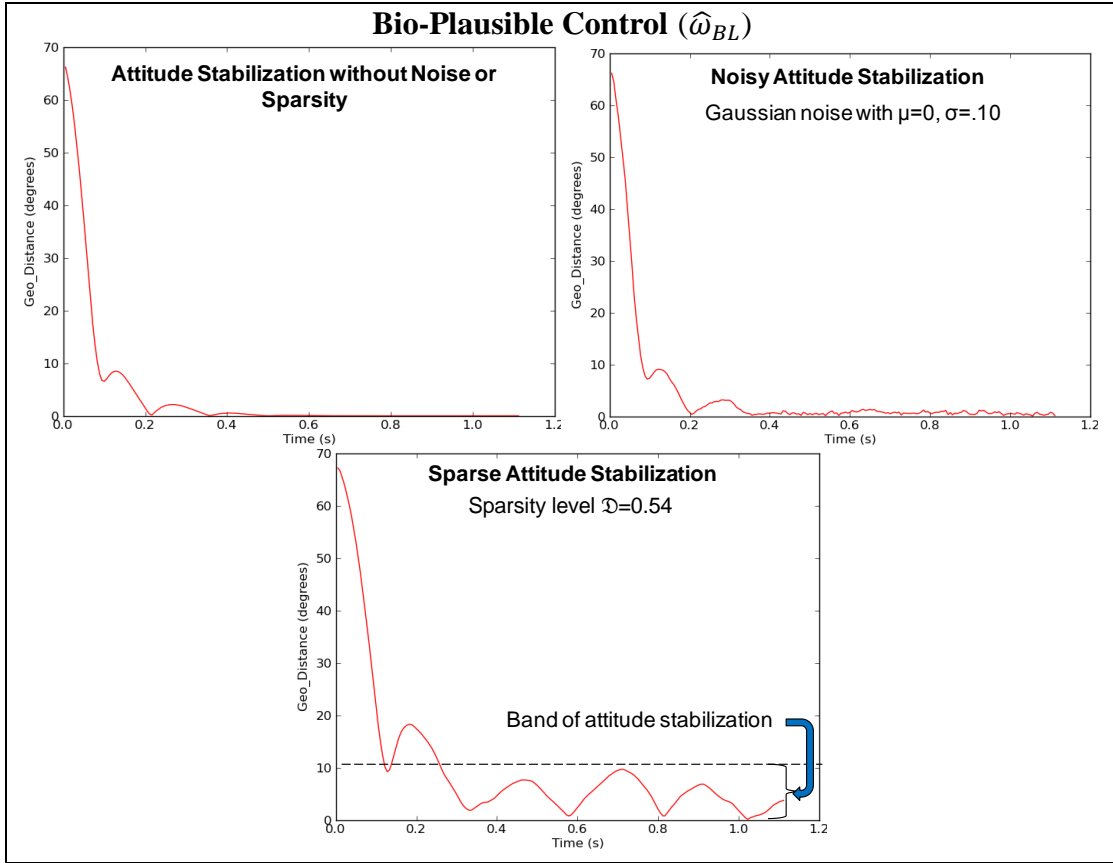


Figure 5. Bio-plausible attitude stabilization in the idealized, noisy, and sparse cases, respectively.

Having established the functionality of  $\hat{\omega}_{BL}$ , in a noise-free environment, we next investigated the algorithm's efficiency for control in the presence of increasing noise, a parameter that correlates directly with sensor quality.

As previously mentioned,  $\hat{\omega}_{BL}$  may be arbitrarily inaccurate in terms of scale. Therefore, a comparison of the least squares and bilinear estimates requires different gains to achieve attitude stabilization than the more physically exact  $\hat{\omega}_{LS}$ . Fortunately, this scale problem does not render use of  $\hat{\omega}_{BL}$  ineffective.

To provide valid noise components in the simulations, we tailored a standard webcam to render the ommatidia view provided by the GUF and used it to measure the standard deviation of luminance in a stationary image collected in an appropriate interior environment, shown in figure 4b. Using this as a reference point, we then added Gaussian noise with a range of standard deviations to the simulated luminance values of the GUF environment. This noise is added to take away from the deterministic nature of our fly moving through the pre-rendered 3-D domain and more accurately simulate real-world sensors.

Simulations of the GUF with greater motion degrees of freedom illuminated an important relationship: the less restricted the fly’s path, the smaller the range of gains for which stabilization occurs. When we limited motion to yaw, the system stabilized over a comparatively wide range of gains; for full attitude, the gains’ range of stability narrowed sharply. Plots of attitude stabilization using our described bio-plausible simulation method are shown in figure 5. As expected, the deterministic “fully informed” system stabilized within a narrower spatial band than its noisy or sparse counterparts, when all other simulation parameters were held constant.

As we previously described, the GUF visualization environment and a rigid body physics model were used. In the simulations presented here, we fixed the goal image, the starting image (or initial orientation data), and the reaction rate constants across differing levels of sparsity and noise. The sparsity plot of figure 5 shows that, as the decision time is shortened and the number of computations performed is reduced, the drift about the goal image becomes greater. In the Slow Computing SSA algorithm, after each time step the system defaults to zero. It then samples a certain number of pixels from the visual field for that time step and outputs a corresponding control torque. When we modified the computation scheme so that the control torque depends both on information accumulated instantaneously and that which is available from the last updated time (the system does not reset memory to zero), we observed that attitude stabilization does not occur. We postulate that the presence of outdated information confounds the effects of information sampled from the present. Thus, we determined that it is better to only actuate the system based on instantaneous visual information.

The accumulation of information from each ommatidium in the slow computing architecture is essential to optimizing effective control decisions while constraining power use. For all simulations comparing noise and sparsity, the rate constants are held uniformly constant. Some examples of uniformity and departures from uniformity of rate constants are illustrated in figure 6. In our model, the geometric distribution of ommatidia information accumulation in the fly eye was coerced to be a square for interpretive simplicity, where each half of the square corresponds to the respective half of the fly’s compound vision. The pixels represent

computational elements operating in parallel. The brightness of each pixel is equivalent to the number of times a particular element has activated (or “fired”). For all plots shown, information is accumulated independently. In the third plot of figure 6, we provide an example of a departure from uniform propensity of the computational elements to provide system information. In reality, most biological systems do not have neurons or sensors functioning uniformly. Thus uniformity of reaction rates is not a valid assumption. Additionally, the accumulation of information across sensor networks may follow complex connectivity dependencies.

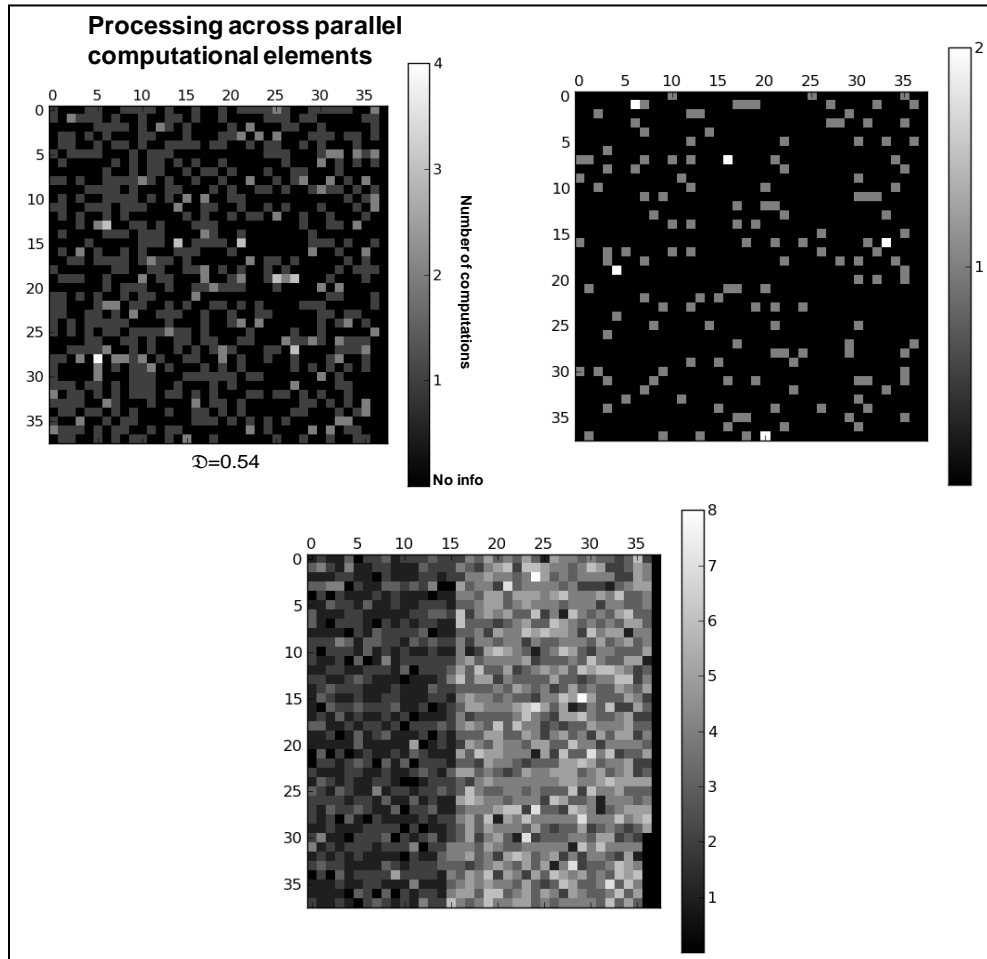


Figure 6. Examples of varying levels of information sparsity and propensities for computational elements to provide visual control information.

In this initial treatment, we have opted for a homogenous and independent network (via the choice of the identity matrix for the stoichiometric matrix). Many potential challenges arising from network complexity may be ignored for the sake of conventional circuit design, but as circuits become more effective at mimicking biology, these complexities will need to be fully considered. Network complexity is finding its way into electronic hardware via neuromorphic

integrated circuit chips, a technology that will be useful in the design of future insect-scale robotics and power-constrained distributed sensor networks such as wearable electroencephalography (EEG) systems.

We note that the data presented in figure 7 are preliminary, with only three trials conducted at each simulation setup. Extensive computation time of the slow computing simulation limited our ability to run enough trials to have a strong understanding of the boundary conditions in the time available for the project, but analysis using more trials is planned. In the ideal case where all visual information is incorporated into the control signal and the sensor quality is perfect, the geodesic degree difference between the goal image and the present orientation of the system nears zero. One would expect that increasing sparsity of visual information and lower quality sensor input would lead to a more adulterated control signal. This relationship is demonstrated in figure 7. We define the bound of geodesic distance/degrees between the instantaneous system orientation and the goal orientation as the band of stabilization. A schematic of the relationship between the fly view, band of stabilization, and the bound on the geodesic distance between present and goal orientation is also given. A narrower band of stabilization implies the system remains relatively close to the goal image (orientation) while a broader band means an increased drifting about the goal image. When there is no bound on this geodesic distance, the system is said to be unstable. The simulation results shown in figure 7 demonstrate that as the quality of the sensor, i.e., the noise in the luminance values, decreases, the band of stabilization increases. A similar but sharper relationship is present for sparsity of visual information: as sparsity increases, the band of stabilization increases faster and stability is lost beyond a certain point. The interaction of sparsity, noise, and band of attitude stabilization produces boundary conditions whose analytical properties need to be explored but very well may be intractable. At a minimum, these results give us an intuitive idea of the amount of noise and sparsity with which the system can achieve some level of stabilization.

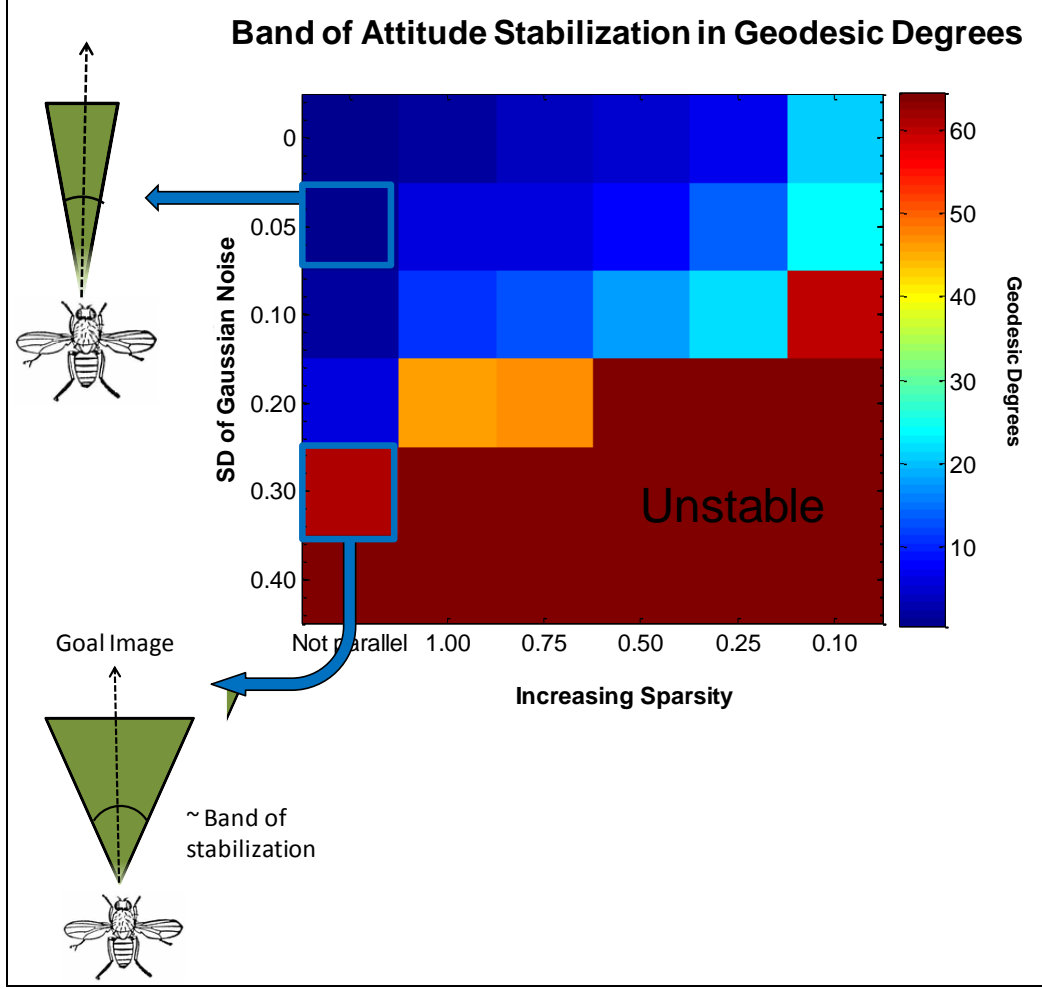


Figure 7. Map of stability control using the Slow Computing algorithm, as a function of noise and information sparsity. Note that noisier and sparser visual information degrades the control signal to a point where the system cannot stabilize.

## 5. Conclusions

In achieving simulated attitude stabilization, many system inputs must be balanced. As an example, table 3 shows the complexity of engineering an effective autonomous robotic insect. In each category, there is an optimal range that may enable more flexibility in the other parameters of the system. To illustrate this point, we note that within a more manageable amount of noise, for example, a 5% standard deviation as observed in figure 7, the system stabilizes over a wider range of gains. Similarly, with a more effective visual sampling scheme, the system could achieve attitude stabilization from greater initial geodesic distance. Optimizing each category's parameters requires distilling complex nonlinearities of the system to desirable ranges. These optimization problems are beyond the scope of this report. Instead, we have relied on

serendipitous discoveries in simulation that motivate the development of rigorous explanations in the future.

Table 3. Different parameters that have an effect on attitude stabilization.

Parameter Space			
Physics	Control	Parallelization	Visual Sensor Information
<ul style="list-style-type: none"> <li>• Inertial Matrix</li> </ul>	<ul style="list-style-type: none"> <li>• Gains</li> <li>• Initial Geodesic Distance from Goal Image</li> </ul>	<ul style="list-style-type: none"> <li>• Network Structure</li> <li>• Rate Constants</li> <li>• Sparsity</li> </ul>	<ul style="list-style-type: none"> <li>• Noise</li> <li>• Orientation</li> </ul>

Bio-plausible control has proved resilient to the difficulties of slow computing. That is, with sparse visual input accumulated in a random way, the control system still behaves relatively well for the rigid-body physics model considered. In simulation, low power processing of visual input produced effective stabilization for a range of noise and sparsity. As other bio-plausible (linear and parallelizable) control laws are developed, it will become necessary to test their efficacy on slow computing systems. As multiple behavioral controls become vetted on asynchronous parallel processing, we may see them implemented on low power systems with confidence.

---

## 6. Further Directions

### 6.1 Improving the Physics

The rigid-body physics model presently in use provides a decent approximation; however, more accurate physical descriptions of *Drosophila melanogaster* are available. We may instead model the physics of *Drosophila* flight as a compound rigid body with wing aerodynamics. This physics engine is described in Dickson et al. (1); it may be found in the *fmech* component of the GUF, and it is anticipated that it will lend additional accuracy to our simulation. The closest physics engine to actual *Drosophila* flight would be based on *computational fluid dynamics* (CFD).

### 6.2 Contrast Relationship

It is important to recognize that in the PD controller, the desirable range of gains is those that produce system stabilization. Particular relationships between the gains and attitude stabilization depend on the physical system at hand and contrast of the visual input. The controller would be more adaptable to variation in image contrast if the gains were functions of the contrast rather



than simple constants, but such functions would require image-wide computation and may be not parallelizable. In low-contrast environments, biological systems would intuitively be more likely to place greater emphasis on other sensory modalities.

### 6.2.1 A Probability Density Function for $\mathfrak{D}$

A probability distribution for *Drosophila melanogaster* behavioral dynamics has been developed in previous investigations by Sorribes et al. (21). In particular, the erratic and “bursty” behavior is formalized as a complimentary Weibull cumulative distribution for the time between fly movements. Thus, an insightful direction for greater biological accuracy would be to vary the control decision time  $\mathfrak{D} \sim \exp\left(-\left(\frac{t}{\lambda}\right)^k\right)$  with the scale parameter  $\lambda = 6.0$  and the shape parameter  $k = 0.45$ , both empirically derived. We would expect to see highly bursty behavior correspond to very low values of  $\mathfrak{D}$ , while more consistent behavior would correspond to longer decision control times, as we have seen in our own simulations so far.

### 6.2.2 Kernel Smoothing

Gaussian kernels are used as a default for image smoothing; however, depending on the situation, other kernels may be more appropriate. As the system moves towards hardware implementation, this topic will need to be explored so that optimal performance is achieved with respect to image processing. We added Gaussian noise to the luminance input, so for our purpose Gaussian kernel smoothing worked well. The smoothing kernel, in principle, should correspond to the noise present in the image. The reader is referred to Tayebi (17) and Wand and Jones (20) for more information.

### 6.2.3 Network Architecture

The sensor geometry also leads to some important questions. Presently, we treat all the sensors as independently influencing the system’s accumulation of information. Instead, the connectivity between sensors could be altered to optimize information gathering across the visual field. For instance, one could imagine if a particular sensor had visual information, the sensors in the “neighborhood” immediately surrounding that sensor to be less likely to ascertain information. This approach could be seen as having the reaction rate constants be functions of the nearby sensor indices. A graph theoretic treatment of the relationship between sensor geometry/connectivity, sparsity of control decision information, and system stabilization would be a novel approach to this problem.

### 6.2.4 Learning Sensor Orientation

Presently, we are assuming a sensor orientation given by the fruit fly *Drosophila melanogaster*. In our calculation scheme, these orientations have been encoded by the matrix  $A$ . Many implementations of the methods here may not be based upon the compound vision of the fruit fly and therefore would have different visual sensor orientations. Learning algorithms exist to establish the sensor orientations in order to design accurate adaptive control systems. A training

signal from an inertial measurement unit (IMU) or haltere may be used as the true value for angular velocity. The system then learns its sensor orientations using visual input and the known value for the signal. Censi et al. (18) and Han et al. (5) provide insight on how this could be implemented.

### 6.2.5 Broadening the Control Domain

In this report, we narrowed our focus to attitude stabilization. Comparable methods have been extended to bio-plausible control designs for pose stabilization (six degrees of freedom of motion). This greater controllability corresponds mathematically to expanding our robotic motion from  $SO(3)$  to  $SE(3)$ , the Special Euclidean group, which corresponds to translations and rotations of objects in  $\mathbb{R}^3$ . Thus, full pose stabilization would be a natural next step. In addition to stabilization, navigation across complex pathways with obstructions will be the ideal end goal.

### 6.2.6 Hardware Implementation

Hardware implementation of bio-plausible control systems presents several simultaneous challenges. The parallelizable control computations would occur on low-grade circuitry in parallel. An FPGA would be well suited to the experimental design of circuitry. With more intricate information networks, neuromorphic chips would become necessary. Small unstable flying platforms currently require RTK, GPS, or Vicon closed-circuit camera technology in order to achieve effective control. Such processes require significantly more power than low-grade cameras coupled with a PD controller implemented on cheap parallel circuitry. In addition, RTK, GPS, and Vicon require significantly more external resources.

In practice, we may achieve stabilization for small flying systems like quad rotors through visual input alone using our bio-plausible control methods. These visually based methods can respond very quickly due to the parallel computing structure and therefore are more adaptive to varying environmental conditions like gusts of wind. A flying system could be stabilized near its current position through an instantaneous snapshot, which becomes the “goal image” and provides an effective low power method for relatively fast stabilization.

In the long term, we want integrate multiple sensor modalities to produce a more adaptive and robust control system. This goal coincides with exploring how sparse the information may be while producing an accurate system response; the *cross-sensitivity* of sensor modalities is of key interest here. With more diverse sensory input, we would expect the system to be able to stabilize more effectively with sparser information for each individual sensor modality. Such sensor modalities may include vision, haltere input, acoustic sensing, and infrared vision.

The simulation of multiple sensor modalities on the slow computing architecture could be approached as coupled (or “tripled”) continuous-time Markov Chains. The Gillespie SSA could be tailored to handle this experimental setup, although simulating the slow computing

architecture would be very computationally intensive, especially with the appropriate physics engine and translational motion.

Figure 8 summarizes the scope of the future work in design efforts towards developing control of an insect-scale autonomous robotic platform.

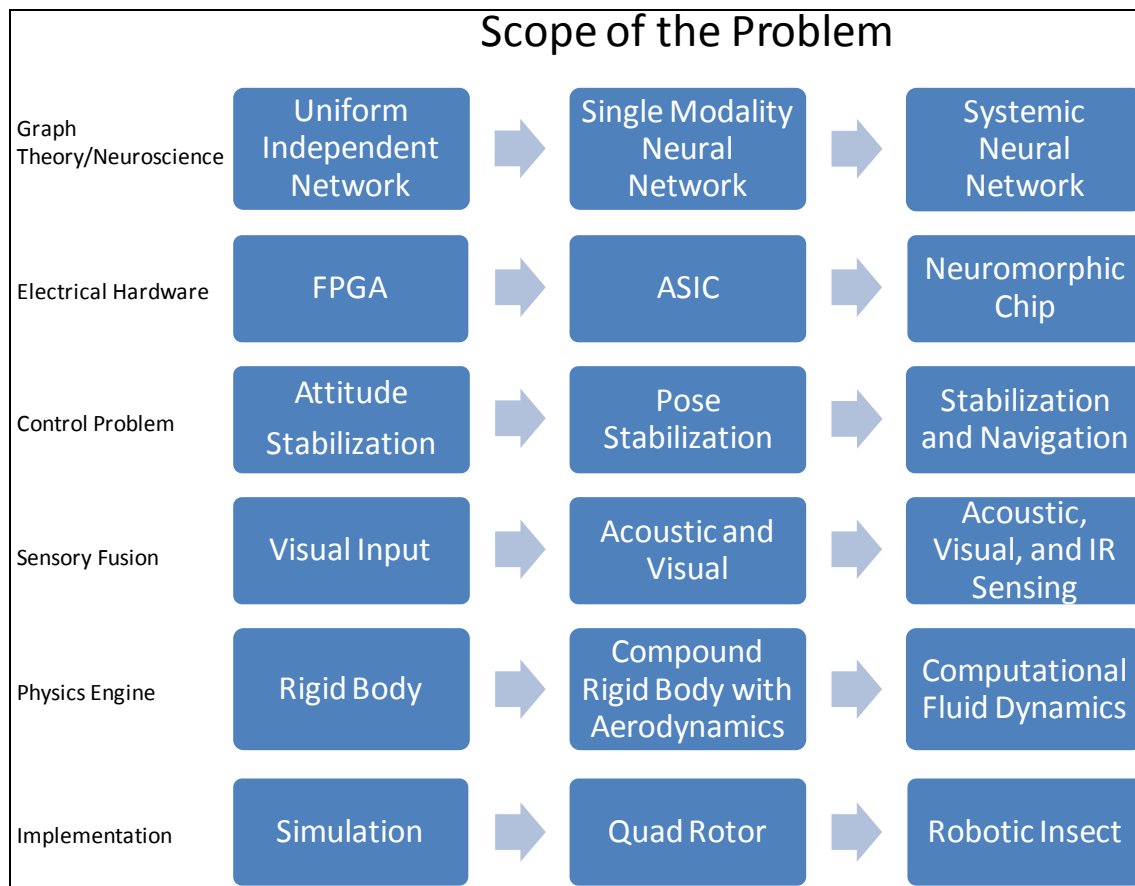


Figure 8. Roadmap of design efforts towards control of an insect-scale autonomous robotic platform.

---

## 7. References

---

1. Dickson, W. B.; Straw, A. D.; Poelma, C.; Dickinson, M. An Integrative Model of Insect Flight Control. in *44th AIAA Aerospace Sciences Meeting and Exhibit*, 2006.
2. Epstien, M.; Waydo, S.; Fuller, S. B.; Dickson, W.; Straw, A.; Dickinson, M. H.; Murray, R. M. Biologically Inspired Feedback Design for Drosophila Flight. in *American Control Conference*, New York, NY, 2007, 3395–3401.
3. Fry, S. N.; Rohrseitz, N.; Straw, A.; Dickinson, M. H. Visual Control of Flight Speed in Drosophila Melanogaster. *J Exp Biol* **2009**, (212), 1120–1130.
4. Han, S.; Censi, A.; Straw, A. D.; Murray, R. M. A Bio-plausible Design for Visual Pose Stabilization. in *IEEE Conference on Intelligent Robots and Systems*, 2010, 5679–5686.
5. Han, S.; Censi, A.; Straw, A. D.; Murray, R. M. A Bootstrappable Bio-plausible Design for Visual Pose Stabilization. in *IEEE International Conference on Robotics and Automation*, 2010, 5679–5686.
6. Gillespie, D. Exact Stochastic Simulation of Coupled Chemical Reactions. *J. Phys. Chem.* **1977**, 81 (25), 2340–2361.
7. Conroy, J.; Gremillion, G.; Ranganathan, B.; Humbert, J. S. Implementation of Wide-field Integration of Optic Flow for Autonomous Quadrotor Navigation. *Autonomous Robots* **Oct. 2009**, 27 (3), 189–198.
8. Bause, F.; Kritzinger, P. *Stochastic Petri Nets: An Introduction to the theory*. Wiesbaden, 2002.
9. Higham, D. Chemical Master Equation and Langevin Regimes for a Gene Transcription Model. in *Conf. on Numerical Analysis of Partial Differential Equations*, vol. 408, Manchester, Nov. 2006.
10. Kammash, M. The Chemical Master Equation in Gene Networks: Complexity and Approaches. Cal-Tech, 2006.
11. Baker, A. *Matrix Groups: An Introduction to Lie Group Theory*; Springer, 2003.
12. Chaturvedi, N. A.; Sanyal, A. K.; Harris McClamroch, N. Rigid Body Attitude Control: Using Rotation Matrices for Continuous, Singularity-free Control Laws. in *IEEE Control Systems Mag. on Decision & Control*, 2006.
13. Diebel, J. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. 2006.

14. Murray, R.; Li, Z.; Sastry, S. *A Mathematical Introduction to Robotic Manipulation*; CRC Press, 1994.
15. Stuelpnagel, J. On the Parameterization of the Three-dimensional Rotation Group. *SIAM Review* **Oct. 1964**, 6 (4), 422–430.
16. Adams, J. F. Vector Fields on Spheres. *Ann. of Math* **1962**, (75), 603–632.
17. Tayebi, A. Unit Quaternion Observer Based Attitude Stabilization of a Rigid Spacecraft Without Velocity Measurement. in *IEEE Conf. on Decision & Control*, 2011.
18. Censi, A.; Han, S. A Bio-plausible Design for Visual Attitude Stabilization. in *48th IEEE Conference on Decision and Control*, 2009.
19. Simoncelli, E. P.; Adelson, E. H.; Heeger, D. J. Probability Distributions of Optical Flow. in *IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, 1991.
20. Wand, M. P.; Jones, M. C. Kernel Smoothing. *Monographs on Statistics and Applied Probability* **1995**, 60.
21. Sorribes, A.; Armendariz, B.; Lopez-Pigozzi, G. D.; Murga, C.; Polavieja, G. G. The Origin of Behavioral Bursts in Decision-Making Circuitry. *PLoS Comput. Bio* **2011**, 7 (6).

---

## List of Symbols, Abbreviations, and Acronyms

---

3-D	three-dimensional
ASICs	application-specific integrated circuits
CFD	computational fluid dynamics
CME	Chemical Master Equation
EEG	electroencephalography
FPGAs	field-programmable gate arrays
GPS	global positioning system
GUF	Grand Unified Fly
IMU	inertial measurement unit
PD	proportional/derivative
RTK	real-time kinetic
SSA	Stochastic Simulation Algorithm

NO. OF COPIES	ORAGANIZATION
1 ELEC	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
2	CALIFORNIA INSTITUTE OF TECHNOLOGY ATTN MC 107-81 R MURRAY ATTN MC 136-93 S HAN 1200 E CALIFORNIA BLVD PASADENA CA 91125
1	UNIVERSITY OF MARYLAND AUTONOMOUS VEHICLE LABORATORY ATTN S HUMBERT 3139 JEONG H KIM ENGINEERING BLDG BLDG 225 COLLEGE PARK MD 20742
1	US ARMY RSRCH LAB ATTN RDRL VTU V C KRONINGER ABERDEEN PROVING GROUND MD 21005
18	US ARMY RSRCH LAB ATTN IMNE ALC HRR MAIL & RECORDS MGMT ATTN RDRL CII A E STUMP ATTN RDRL CII A J FINK ATTN RDRL CII A S H YOUNG ATTN RDRL CIN B SADLER ATTN RDRL CIN T P YU ATTN RDRL CIO LL TECHL LIB ATTN RDRL CIO LT TECHL PUB ATTN RDRL SED E B MORGAN ATTN RDRL SED E S BEDAIR ATTN RDRL SER E R PROIE ATTN RDRL SER L A WICKENDEN ATTN RDRL SER L B PIEKARSKI ATTN RDRL SER L J PULSKAMP ATTN RDRL SER L R POLCAWICH ATTN RDRL SER L V GANESAN ATTN RDRL SER L W NOTHWANG ATTN RDRL SER U G SMITH ADELPHI MD 20783-1197

TOTAL: 23 (1, 22 HCS)

INTENTIONALLY LEFT BLANK.